8x930Ax (8x930AD, 8x930AE) SPECIFICATION UPDATE

Release Date: February 1999

Order Number: 272940-017

Notice: The 8x930Ax may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are documented in this Specification Update.

Contents

REVISION HISTORY (Sheet 1 of 2)

Rev. Date	Version	Description
08/30/96	001	This is the new Specification Update document.
09/17/96	002	Added erratum number 9609001. Modified workaround for specification clarification number 001. Added document changes 001 and 002.
09/20/96	003	Added erratum number 9609002.
10/04/96	004	Added errata numbers 9610001, 9610002, 9610003, 9610004. Added A-2 stepping information.
11/13/96	005	Added errata numbers 9611001 and 9611002. Added A-1, A-2, and A-3 stepping information.
12/04/96	006	Added errata numbers 9612001 and 9612002. Corrected the errata table to indicate that five of the errata items were fixed in the A3 stepping: 9609001, 9610001, 9610002, 9610003, and 9610004. Added specification clarification 002. Deleted document changes 001 and 002. The changes were implemented in the datasheet (Order Number 272917-002). Added document changes 003, 004, 005, 006, 007, and 008.
1/8/97	007	Changed status of 9609002, 9610003, 9611002, 9612001 and 9612002 to Fixed. Changed status of 9611001 to No Fix. Added specification changes 001, 002, and 003. Updated specification clarification 002. Added documentation changes 009 and 010. Added information for stepping A-4.
2/5/97	008	Added documentation changes 011, 012, 013, and 014.
4/4/97	009	Added errata numbers 9704001 and 9704002. Added specification clarification 003.
5/9/97	010	Added errata numbers 9705001 and 9705002.

REVISION HISTORY (Continued) (Sheet 2 of 2)

Rev. Date	Version	Description
7/9/97	011	Added firmware workaround to erratum number 9705002. Added erratum 9706001. Added specification clarification 004.
11/4/97	012	Added errata numbers 9711001, 9711002, and 9711003.
4/3/98	013	Added erratum 9804001. Added specification clarification 003.
6/1/98	014	Added erratum 9806001. Added specification changes 004 and 005.
7/6/98	015	Added specification change 006.
8/3/98	016	Added document change 015.
2/5/99	017	Added Workaround #2 to erratum 9612001.

PREFACE

As of July, 1996, Intel has consolidated available historical device and documentation errata into this document type called the Specification Update. We have endeavored to include all documented errata in the consolidation process, however, we make no representations or warranties concerning the completeness of the Specification Update.

This document is an update to the specifications contained in the Affected Documents/Related Documents table below. This document is a compilation of device and documentation errata, specification clarifications and changes. It is intended for hardware system manufacturers and software developers of applications, operating systems, or tools.

Information types defined in Nomenclature are consolidated into the specification update and are no longer published in other documents.

This document may also contain information that was not previously published.

Affected Documents/Related Documents

Title	Order
8X930Ax Universal Serial Bus Peripheral Controller Data sheet	272917-003
8x930Ax, 8x930Hx Universal Serial Bus Microcontroller User's Manual	272949-001

Nomenclature

Errata are design defects or errors. These may cause the published (component, board, system) behavior to deviate from published specifications. Hardware and software designed to be used with any component, board, and system must consider all errata documented.

Specification Changes are modifications to the current published specifications. These changes will be incorporated in any new release of the specification.

Specification Clarifications describe a specification in greater detail or further highlight a specification's impact to a complex design situation. These clarifications will be incorporated in any new release of the specification.

Documentation Changes include typos, errors, or omissions from the current published specifications. These will be incorporated in any new release of the specification.

NOTE:

Errata remain in the specification update throughout the product's lifecycle, or until a particular stepping is no longer commercially available. Under these circumstances, errata removed from the specification update are archived and available upon request. Specification changes, specification clarifications and documentation changes are removed from the specification update when the appropriate changes are made to the appropriate product specification or user documentation (datasheets, manuals, etc.).

SUMMARY TABLE OF CHANGES

The following table indicates the errata, specification changes, specification clarifications, or documentation changes which apply to the 8*x*930A*x* product. Intel may fix some of the errata in a future stepping of the component, and account for the other outstanding issues through documentation or specification changes as noted. This table uses the following notations:

Codes Used in Summary Table

Steps

X: Errata exists in the stepping indicated. Specification Change or Clarification that applies to this stepping.

(No mark)
or (Blank box): This erratum is fixed in listed stepping or specification change does not apply to listed stepping.

Page

(Page): Page location of item in this document.

Status

Doc: Document change or update will be implemented.

Fix: This erratum is intended to be fixed in a future step of the component.

Fixed: This erratum has been previously fixed.

NoFix: There are no plans to fix this erratum.

Eval: Plans to fix this erratum are under evaluation.

Row

|

Change bar to left of table row indicates this erratum is either new or modified from the previous version of the document.

Errata (Sheet 1 of 2)

Number	Steppings				Page	Status	ERRATA
	A1	A2	A3	A4			
9609001	X	X			15	Fixed	The V_{OH} of the 8*x*930A*x* A-1and A-2 Stepping Does Not Meet the V_{OH} DC Specification Published in the Data Sheet
9609002	X				15	Fixed	8*x*930A*x* Transmit FIFO Underrun During an Isochronous Transfer Does Not Set the TXERR Bit in TXSTAT
9610001	X				15	Fixed	Powerdown Wake-up with External Reset
9610002	X	X			16	Fixed	Occasional Timeout by Function Endpoint
9610003	X	X			16	Fixed	SOF# Pin Pulses When the Micro-controller Leaves Suspend Mode
9610004	X	X			16	Fixed	CRC16 Error on a Data Packet While the Endpoint's Receive FIFO Is Not Ready
9611001	X	X	X	X	17	No Fix	Clearing RXSETUP Bit When Transmit FIFO Data Register Is Not Empty In Low-clock Mode
9611002	X	X			17	Fixed	Low Speed Functionalities Not Guaranteed Below 4.5 V
9612001	X	X	X		18	Fixed	Receive FIFO RXFFRC Error
9612002	X	X			21	Fixed	Low-speed Signaling Marginal on Some Devices
9704001	X	X	X	X	21	Fix	Low-speed USB VCRS Electrical Characteristics Marginality
9704002	X	X	X	X	21	Fix	Low-speed Remote Wakeup Function
9705001	X	X	X	X	22	Fix	Serial Port Auto Address Recognition Errata
9705002	X	X	X	X	22	Fix	Isochronous Transfer RXCNT Errata
9706001	X	X	X	X	24	Fix	Timer 2 Idle Mode Wake Up Errata

Errata (Sheet 2 of 2)

Number	Steppings				Page	Status	ERRATA
	A1	A2	A3	A4			
9711001	X	X	X	X	25	Fix	Incorrect Response On Receive FIFO Overflow in Low Clock Mode
9711002	X	X	X	X	25	Fix	External Interrupt Flags are Not Set when the TCON SFR is Simulta-neously Changed
9711003	X	X	X	X	26	Fix	PCA Capture Flag is Not Set when a Capture Occurs with PCA Timer Overflow
9804001	X	X	X	X	26	Fix	Interrupts of 3 or More Priority Levels Occuring at the Same Time
9806001	X	X	X	X	28	No Fix	8x930 Cold Boot Issue On Some Hosts

Specification Changes

Number	Steppings				Page	Status	SPECIFICATION CHANGES
	A1	A2	A3	A4			
001			X	X	30	Doc	V_{OH} When I_{OH} = -60 µA
002			X	X	30	Doc	Extended Data Float Option
003			X	X	30	Doc	AC Characteristics Changed
004	X	X	X	X	30	Doc	Series Resistor Requirement for Impedance Matching Changed to 22 Ohms (no longer 27 – 33 ohms)
005	X	X	X	X	30	Doc	Minimum Operating Voltage (Vcc) Specification for the 8x930Ax
006	X	X	X	X	30	Doc	8x930xx Devices No Longer Supported for Low Speed

Specification Clarifications

Number	Steppings				Page	Status	SPECIFICATION CLARIFICATIONS
	A1	A2	A3	A4			
001	X	X			32	Doc	T_{RHDZ1} Timing
002			X	X	35	Doc	Extended Data Float Option
003	X	X	X	X	38	Doc	ECAP Pin Usage to Supply 3.0v to 3.6v Voltage for 1.5K Ohm USB Pullup Terminator
004	X	X	X	X	39	Doc	TXCNT Must Be Written With Correct Byte Count
005	X	X	X		20	Doc	Receive FIFO RXFFRC Error (Errata)

Documentation Changes

Number	Document Revision	Page	Status	DOCUMENTATION CHANGES
003	001	40	Doc	Nonvolatile Memory Verification Port Labeled Incorrectly
004	001	40	Doc	Incorrect Address Given for TXSTAT SFR
005	001	40	Doc	Incorrect Signature Byte
006	001	40	Doc	Power-on Reset Capacitor Value Changed
007	001	40	Doc	SCON SFR’s REN Bit Description Incorrect
008	001	40	Doc	Extraneous Footnote in RXCON SFR
009	001	41	Doc	Power Off Flag Voltage Values
010	001	41	Doc	W_{CLK} Description Incorrect
011	001	41	Doc	Configuration Byte Misspelled
012	001	41	Doc	RTWCE Description Inaccurate
013	001	41	Doc	RL Instruction Misspelled
014	001	41	Doc	Footnote Incorrect in Data Instructions Table
015	001	13	Doc	Capacitor Value Change in Figure 3 of this Specification Update

IDENTIFICATION INFORMATION

Markings

Product	Part Number	Stepping	Marking	Comment
8x930Ax Step A1	N80930AD	A1	Q 866	sample QDF
	N80930AE		Q 892	sample QDF
	N83930AE		R xxxx	
8x930Ax Step A2	N80930AD2	A2	no marking	shipping media = tubes
			Q 873	sample QDF
			S L23D	shipping media = tape ‘n’ reel
	N83930AD2		R xxxx	
	N83930AE2		R xxxx	
8x930Ax Step A3	N80930AD3	A3	no marking	ship media = tubes
			Q 873	sample QDF
			S L24E	shipping media = tape ‘n’ reel
	N83930AD3		R xxxx	
	N83930AE3		R xxxx	
8x930Ax Step A4	N80930AD4	A4	no marking	ship media = tubes
			Q 802	sample QDF
			S L26M	shipping media = tape ‘n’ reel
	N83930AD4		R xxxx	
	N83930AE4		R xxxx	

ERRATA

9609001. The V_{OH} of the 8x930Ax A-1and A-2 Stepping Does Not Meet the V_{OH} DC Specification Published in the Data Sheet

PROBLEM: When Port 1, 2, and 3 are in quasi-bidirectional mode, their V_{OH}s are below the target specification as below:

$V_{OH} = V_{CC}$ - 0.8 V (instead of V_{CC} - 0.3 V) @10 µA
$V_{OH} = V_{CC}$ - 1.7 V (instead of V_{CC} - 0.7 V) @30 µA
$V_{OH} = V_{CC}$ - 2.7 V (instead of V_{CC} - 1.5 V) @60 µA

IMPLICATION: The fanout of port 1, 2, and 3 pins are reduced.

WORKAROUND: External buffers can be used to provide the required drive capability needed for interfaces that require more drive than the 8*x*930A*x* can support.

STATUS: Fixed. Refer to Summary Table of Changes for affected steppings.

9609002. 8x930Ax Transmit FIFO Underrun During an Isochronous Transfer Does Not Set the TXERR Bit in TXSTAT

PROBLEM: A transmit FIFO underrun in the 8x930Ax during an isochronous transmission does not set the TXERR bit in the TXSTAT special function register (SFR). As a result, the 8x930Ax remains in an infinite IN-token state and responds with an invalid packet identification (PID) of FFH for subsequent IN-tokens.

IMPLICATION: In this state, the 8*x*930A*x* will not respond correctly to host USB commands.

WORKAROUND: Firmware must avoid causing a FIFO underrun. If a FIFO underrun occurs, the firmware can clear the FIFO by setting the TXCLR bit in the TXCON special-function register (SFR) to recover from the infinite IN-token state.

STATUS: Fixed. Refer to Summary Table of Changes for affected steppings.

9610001. Powerdown Wake-up with External Reset

PROBLEM: If an external reset is applied to wake up the microcontroller from powerdown mode, the instruction following the powerdown instruction may be executed before the branch to the reset vector occurs.

IMPLICATION: The instruction executed may corrupt a memory location (RAM, registers, or SFRs). This is an issue for systems that require data retention in memory after a reset.

WORKAROUND: Add a no-operation (NOP) instruction after the powerdown instruction to prevent memory corruption.

STATUS: Fixed. Refer to Summary Table of Changes for affected steppings.

9610002. Occasional Timeout by Function Endpoint

PROBLEM: The function occasionally times out (stops communicating with the USB host) if the LC bit of the PCON SFR is cleared frequently.

IMPLICATION: The function loses communication with the host occasionally.

WORKAROUND: None. Firmware should check the LC bit and, if it is already clear, firmware should not clear it again.

STATUS: Fixed. Refer to Summary Table of Changes for affected steppings.

9610003. SOF# Pin Pulses When the Microcontroller Leaves Suspend Mode

PROBLEM: When a resume or a remote wake-up causes the microcontroller to exit suspend mode, a pulse occurs on the SOF# pin.

IMPLICATION: An invalid SOF# pulse is generated. An application that uses the SOF# pulse may see an invalid SOF# time stamp.

WORKAROUND: None.

STATUS: Fixed. Refer to Summary Table of Changes for affected steppings.

9610004. CRC16 Error on a Data Packet While the Endpoint's Receive FIFO Is Not Ready

PROBLEM: When a CRC error on a data packet (from the host) appears on an endpoint while the endpoint's receive FIFO is not ready, subsequent host transactions are ignored. (The "not ready" condition can be the result of a FIFO underrun, a FIFO overflow, or the FIFO's RXFLG register's RXFIF1:0 bits = '11'.) Any IN tokens to that endpoint will time out. The endpoint will respond only to OUT tokens.

IMPLICATION: The endpoint will be unable to respond correctly to subsequent IN tokens.

WORKAROUND: None.

STATUS: Fixed. Refer to Summary Table of Changes for affected steppings.

9611001. Clearing RXSETUP Bit When Transmit FIFO Data Register Is Not Empty In Low-clock Mode

PROBLEM: For control endpoint only, when the following conditions below are true, clearing the RXSETUP bit in the RXSTAT Special Function Register (SFR) may put the device in a continuous NAKing state unless the Transmit FIFO data register is flushed (by setting the TXCLR bit in TXCON SFR).

1. LC bit in PCON Special Function Register (SFR) is SET, that is, the CPU core and peripherals is running at 3MHz, specifically for bus-powered applications,
2. The Transmit FIF0 data register is not empty, that is, if the TXFIF1 and TXFIF0 bits in TXFLG SFR is not 00, and
3. The device is continuously NAKing the host's IN token.

This problem is seen on the USB OHCI system.

IMPLICATION: The device will not be able to respond to host command and, host will time-out and the device will not be served by the host.

LC mode is used during bus enumeration by the bus-powered application, thus the self-powered application will not see this issue if the LC bit is cleared immediately after device reset.

WORKAROUND: To avoid the device going into continuous NAKing state, the transmit FIFO data register must be empty, that is, the TXFIF1:0 bits in the TXFLG special function register (SFR) are “00”, when the RXSETUP bit is cleared.

STATUS: No Fix. Refer to Summary Table of Changes for affected steppings.

9611002. Low Speed Functionalities Not Guaranteed Below 4.5 V

PROBLEM: The low speed functionalities of the 8*x*930A*x* (A-1 and A-2) are not tested below the operating voltage of 4.5v. Therefore, these low-speed functions are not guaranteed below the operating voltage of 4.5v.

IMPLICATION: The A-1 and A-2 parts cannot be used in low-speed, bus-powered devices where the supply voltage at the parts is less than 4.5 V.

WORKAROUND: No workaround. This is not an issue with full speed operation.

STATUS: Fixed. Refer to Summary Table of Changes for affected steppings.

9612001. Receive FIFO RXFFRC Error

PROBLEM: After the RXFFRC bit of the RXCON register is set, the RXFIF1:0 bits in the RXFLG register remain “11”. According to specifications, RXFIF1:0 should immediately decrement when RXFFRC is set. This problem occurs when the following conditions are simultaneously met:

1. The function interface unit (FIU) and serial bus interface engine (SIE) write a byte count to the RXCNTL SFR of endpoint 1’s receive FIFO.
2. The CPU sets the RXFFRC bit for any other endpoint’s receive FIFO (not endpoint 1).
3. The receive FIFO for the endpoint in (2) has RXFIF1:0 bits = “11”.

This problem is most likely to occur in low-clock mode when the device has a high data receive rate (bulk mode) on endpoint 1 and one or more of the other receive FIFO endpoints.

IMPLICATION: When the problem occurs, having RXFIF1:0 remain as “11” will cause firmware to incorrectly assume that there are two packets left in the receive FIFO (in dual packet mode), when in reality there is only one packet left. If firmware attempts to read the non-existent second packet, hardware will set the RXURF bit. When the RXURF bit gets set, the 8*x*930A*x* continues to NAK all OUT packets on the affected FIFO. At this point, the FIFO will be in an unknown state, requiring firmware to reset/clear that FIFO.

WORKAROUND: #1 (if you experience 1/3 data loss with this workaround, try workaround #2) Additional code must be added to the firmware location(s) where a non-endpoint 1 receive FIFO is released. This code must determine if the RXFIF1:0 bits are “11” before and after setting the RXFFRC bit. If this is true, and if the RXSEQ bit has not been toggled by hardware during this time, then the error has occurred. At this point, firmware must attempt to release the RXFIFO again by re-setting the RXFFRC bit. This process must be repeated until the RXFIFO is successfully released.

Insert a firmware routine similar to the example shown in Figure 1 in your code at the point where you release the receive FIFO for all endpoints except endpoint 1. The code must be duplicated for each endpoint (except endpoint 1), replacing the “x” in the example’s code labels with the endpoint number.

```
;*******************************************************
; RXFFRC Firmware Workaround
;*******************************************************
; Note: Registers 11 through 14 are utilized in this example.
; Please be sure to save and restore these registers as needed.

RELEASE_FIFO_x:
          mov     A,      RXFLG
          mov     R12,    RXFLG          ; before
          mov     R13,    RXSTAT

          setb    RXFFRC

          ; if (RXFLG_BEFORE = RXFLG_AFTER)
          ;  then { continue and check if RXFIF="11" }
          ;  else { setting RXFFRC was successful }
          cjne    A,    RXFLG, REL_FIFOx_OK

          ; if we get here R11 has RXFLG before and after - no
          ; change
          mov     R14,    RXSTAT

          ; if (RXFLG_AFTER = "11")
          ;  then { continue and check RXSEQ data toggle }
          ;  else { jump to REL_FIFOx_OK}
          anl     R11, #11000000b          ; check RXFIF after
          cjne    A,   #11000000b,  REL_FIFOx_OK

          ; RXFIF bits are "11"    RXFLG="C0"
          ; if (RXSEQ_BEFORE = RXSEQ_AFTER)
          ;  then
          ;   { no data toggle - set RXFFRC again }
          ;  else
          ;   { data toggle - OK jump to REL_FIFOx_OK}
          anl     R13, #10000000b
          anl     R14, #10000000b
          cmp     R13,   R14
          jne     REL_FIFOx_OK

          ; FIFO errata condition: RXFIF was "11" before & after; &
          ; RXSTAT didn't change
RELEASE_FIFO_x_AGAIN:
          ljmp    RELEASE_FIFO_x

REL_FIFOx_OK:
```

Figure 1. RXFFRC Firmware Workaround

WORKAROUND: #2 Make sure that if workaround #1 was attempted, that you backout of the workaround. Then, set RXFFRC as usual but ignore RXCNT entirely. Use a read routine of RXDAT that reads it until RXURF is set and then clear RXURF. An example is provided below that can be a subroutine from within the SOF or can be put in-line as well. The following example is a code suggestion only and responsibility of verification of the workaround lies with the customer as each implementation and application will be different. Also note that this workaround may cause issues with performance sensitive applications and some fine tuning or simliar algorithms may need to be engineered.

```
READ_RXDATA_ROUTINE:
; push registers here as required by your application

moredata:
; read RXDAT in scratch buffers
mov R3, RXDAT
mov R2, RXDAT

; if RXURF then we've read past the end of the FIFO
jb RXURF, nomoredata

; good data, move to R5 and R4 (or wherever you want the data)
mov R5, R3
mov R4, R2

; try to read more data
sjmp moredata

nomoredata:
clr RXURF
; with this setup you will always read until RXURF so it always needs to be
; cleared every time

; now pop whatever registers you pushed and return

ret
```

STATUS: Fixed. Refer to Summary Table of Changes for affected steppings.

9612002. Low-speed Signaling Marginal on Some Devices

PROBLEM: When some A1- and A2-stepping devices are used at low speed (1.5 Mbps), the signals driven out of the transceiver (D_{P0} and D_{M0}) may have longer rise and fall times. As cable length increases, rise and fall times increase, and the crossover point between D_{P0} and D_{M0} decreases. As this crossover point decreases, the data is less likely to be interpreted correctly by the host PC.

IMPLICATION: This problem will cause the host to not recognize signals sent by the device. The host will then terminate communication with the device.

WORKAROUND: There is no workaround, although screening samples may eliminate those devices that exhibit marginal low-speed signaling.

STATUS: Fixed. Refer to Summary Table of Changes for affected steppings.

9704001. Low-speed USB VCRS Electrical Characteristics Marginality

PROBLEM: When performing as a low-speed USB function, the device shows marginality on passing the output signal crossover voltage (VCRS) electrical characteristics/bus timing (refer to section 7.3.2, Bus Timing/Electrical Characteristics, of the ***Universal Serial Bus Specification***, Rev. 1.0).

IMPLICATION: This problem can only affect the device when performing as a low-speed function. This problem occasionally causes the host PC to fail to recognize signals sent by the device, resulting in the termination of communication with the device.

WORKAROUND: Customers who would like to use the device for a low-speed USB function should order the device through order number: N, 80930AD4, S L2FH. These devices pass the specified electrical characteristics.

STATUS: Fix. Please refer to Summary Table of Changes for a list of affected steppings.

9704002. Low-speed Remote Wakeup Function

PROBLEM: When performing as a low-speed USB function and at a cold temperature, the device does not generate resume signalling when instructed by firmware for remote wakeup functionality (done by setting the RWU bit of the PCON1 SFR).

NOTE:

As a full-speed USB function, the remote wakeup feature is functional when the device performs throughout the specified temperature range.

IMPLICATION: When performing as a low-speed USB function, the device cannot support remote wakeup functionality. Note that the remote wakeup feature is functional when the device operates as a full-speed USB function. This remote wakeup function is optional under the ***Universal Serial Bus Specification***, Rev. 1.0.

WORKAROUND: None.

STATUS: Fix. Please refer to Summary Table of Changes for a list of affected steppings.

9705001. Serial Port Auto Address Recognition Errata

PROBLEM: When the 8*x*930A*x* serial port is configured to use auto address recognition feature in multiprocessor communication by setting the SM2 bit in SCON register, the 8*x*930A*x* cannot automatically recognize the sent slave (given) address and broadcast address. This applies to serial port modes 1, 2, and 3.

IMPLICATION: When configured to use this feature, the serial port cannot recognize automatically the given address and broadcast address sent to it. As such, the serial port receive interrupt, RI flag in SCON register will not be set, and no interrupt will be generated. This errata is not applicable to serial port mode 0 as it does not support this feature.

WORKAROUND: Manually check the address if multiprocessor communication, using serial port, is needed. To avoid enabling the auto address recognition feature, do not set the SM2 bit in SCON register. The RI flag is set and interrupt is generated with every serial data received. In the serial port receive interrupt service routine, check the RB8 bit in SCON register to determine if the serial data received is an address byte (RB8=1) or a data byte (RB8=0). If it is the address byte and the address matches with a pre-assigned address, then read the serial data bytes on the subsequent receive interrupts until next address byte is received. If the address byte does not match with the pre-assigned address, then ignore the subsequent serial data bytes received until next address byte is received.

STATUS: Fix. Please refer to Summary Table of Changes for a list of affected steppings.

9705002. Isochronous Transfer RXCNT Errata

PROBLEM: When the 8*x*930A*x* is configured to use dual packet mode in isochronous transfer, the receive FIFO count register (RXCNTx) of the new data packet is corrupted if the ***read completion*** of the previous data packet (setting RXFFRC bit) coincides with ***receive done*** of the new data packet.

WORKAROUND: When this problem occurs, the 8*x*930A*x* will read an incorrect value from the RXCNTx register on the new data packet and hence read an incorrect number of bytes from the receive FIFO. With that, the data read will be either shorter or longer than actually received, the data read will not be correct, and the receive FIFO will overflow or underrun.At the end of an isochronous data read in the start of frame (SOF) ISR, do **not** r

release the RXFIFO by setting the RXFFRC bit. In the very beginning of the next SOF interrupt service routine, before the next OUT token arrives, set the RXFFRC bit to release the RXFIFO that was read in the previous frame. This will prevent the ***read completion*** of the previous data packet (setting RXFFRC bit) from coinciding with the ***receive done*** of the new data packet. An example of a firmware workaround is shown in Figure 2.

STATUS: Fix. Please refer to Summary Table of Changes for a list of affected steppings.

Figure 2. Isochronous Transfer RXCNTx Errata Firmware Workaround Example

```
;*******************************************************************************************************
; Isochronous Transfer Endpoint 1 Initialization Code
; > Use a direct address RAM location (eg. 30h) to store a value for setting or not setting RXFFRC bit:
; > [30h] = 0Ch ->not to set RXFFRC bit,
; > [30h] = 1Ch ->to set RXFFRC bit,
;*******************************************************************************************************
ISO_EP1_INIT:                                   ; example endpoint 1 initialization routine
   mov                                          ; do not set RXFFRC bit when entering SOF ISR for the
   30h, #0Ch                                    ;first time

   ;other Endpoint 1 initialization code
   ret
;*******************************************************************************************************
; START OF FRAME (SOF) Interrupt Service Routine
;  > RXFFRC bit is set to release the isochronous data packet that was read in previous frame
;*******************************************************************************************************
;
SOF_ISR:                                        ; example endpoint 1 isochronous transfer ISR
    push   EPINDEX
   mov    EPINDEX, #01h                         ; select iso endpoint 1
   mov    RXCON, 30h                            ; set or do not set RXFFRC bit depends on value ;in 30
   ; do not add additional instructions before this line; RXFFRC bit must be set before OUT token
   arrives
   push    PSW
   push    PSW1
   push    ACC
   push    R3
   clr     ASOF                                  ;clear SOF interrupt flag
;
EP1_RX_ISOC:                                    ;example ISO receive processing routine
   jb      RXFIF0, EP1_RX_DATA_AVAIL
   jb      RXFIF1, EP1_RX_DATA_AVAIL
   mov     30h, #0Ch                            ; no iso data packet received, don't
                                                ;setRXFFRC bit in next frame
   ljmp    EXIT_SOF_ISR

(Continued)
```

Figure 2. Isochronous Transfer RXCNTx Errata Firmware Workaround Example (Continued)

```
;***************************************************************************************************
; Start of Frame Interrupt Service Routine, Continued
;***************************************************************************************************
;
EP1_RX_DATA_AVAIL:
   mov     30h, #1Ch                             ; to set RXFFRC bit in next frame as it is read in
                                                 ;this frame
   jnb     RXERR, EP1_NO_RX_ERROR
   ljmp    HANDLER_RX1_ERROR_X                   ; jump to iso receive error handling routine
;
EP1_NO_RX_ERROR:
   mov    A,  RXCNTL                             ; get the receive count
   jz   EXIT_SOF_ISR
   mov    R3, A                                  ; temp storage of RXCNTL
;
RX_Loop_x:
   mov     A,   RXDAT                            ; read the iso data received
  ; process the data here, such as save to memory etc
   djnz    R3, RX_LOOP_X
;
EXIT_SOF_ISR:                                    ; restore any registers used
  pop  R3
   pop    ACC
   pop    PSW1
   pop    PSW
   pop    EPINDEX
  reti
```

9706001. Timer 2 Idle Mode Wake Up Errata

PROBLEM: When the 8*x*930A*x* is in idle mode, the interrupt generated by the Timer 2 is n not able to wake up the CPU. The Timer 2 interrupt can be generated in various modes by timer or counter overflow, or by capturing an external high-to-low transition signal on T2EX pin.

IMPLICATION: When this problem occurs, the 8*x*930A*x* will remain in the idle mode. The CPU will not wake up, and the Timer 2 interrupt service routine will not execute.

Note that this erratum is not applicable to powerdown mode operation as all peripherals (including Timer 2) and the CPU stop running in powerdown mode. Powerdown mode is required when the 8*x*930A*x* is suspended by USB to meet the suspend current specification. Idle mode is not.

WORKAROUND: There is no workaround when using Timer 2 to wake up the 8*x*930A*x* CPU from idle mode. If Timer 2 service is needed, use another unused interrupt such as Timer 0 or 1, or External Interrupt 0 or 1. Synchronize it with Timer 2 to wake up the CPU in idle mode and then jump to Timer 2 service routine.

STATUS: Fix. Refer to Summary Table of Changes for affected steppings.

9711001. Incorrect Response On Receive FIFO Overflow in Low Clock Mode

PROBLEM: Intermittently, the device will respond incorrectly to an Out token from the host, which causes an RxFIFO overflow. Instead of timing out and setting the appropriate flags, the device will occasionally ACK the data, signalling a valid reception of data.

1. Device is running in Low Clock Mode (PCON.5 is set).
2. Host incorrectly sends one additional byte to the RxFIFO, that is, the total length sent has to be equal to 1 + physical maximum endpoint size. 17 bytes for Endpoint 0 and 5, 33 bytes for Endpoint 2, 3, and 4, and 1025 bytes for Endpoint 1.

The correct response is a USB timeout.The RxEMP and RxOVF bits in the RXFLG SFR should be set. The RxACK should be cleared and RxSEQ should remain unchanged.

IMPLICATION: The host will see an ACK instead of a timeout and the device is in an unknown state. The host and the device will be out of sequence.

WORKAROUND: No workaround is available. Under normal operation, Endpoints 0 and 5 are configured as an 8-byte FIFO single/dual packet mode, Endpoints 2, 3, and 4 are configured as a 16-byte FIFO single/dual packet mode, and Endpoint 1 should be configured as a 512 byte dual-packet mode if used in ISOC transfer type.

STATUS: Fix. Refer to Summary Table of Changes for affected steppings.

9711002. External Interrupt Flags are Not Set when the TCON SFR is Simultaneously Changed

PROBLEM: The external interrupt flags (IE0 or IE1 bits in TCON SFR) will not be set by the edge-triggered or level-triggered interrupts on the INT0# or INT1# pins when the firmware simultaneously modifies the TCON SFR.

IMPLICATION: No external Interrupt 0 or Interrupt 1 will be generated in this situation.

WORKAROUND: Do not modify the TCON control bits after the timer has been started or the external interrupt bits are enabled. If the control bits require modification when a timer is enabled, Timer2 or PCA timer can be used instead. If two timers are required, and the control bits require modification, a combination of Timer0 and Timer2 or Timer1 and

Timer2 can also be used to ensure that IT0 or IT1are still set when Timer2 control bits in T2CON SFR are modified. If external interrupts are required and the control bits in the TCON SFR require modification, use a level-sensitive type triggered interrupt to ensure that the IE0 or IE1 flags are still set upon clearing/setting of control bits in the TCON SFR.

STATUS: Fix. Refer to Summary Table of Changes for affected steppings.

9711003. PCA Capture Flag is Not Set when a Capture Occurs with PCA Timer Overflow

PROBLEM: When the PCA is used in capture mode, the capture flag (CCFx) in the CCON SFR will not be set upon a positive/negative edge signal on the CEXx pin. This only happens when the PCA timer overflows at exactly the same instance as the positive/negative edge signal on the CEXx pin.

IMPLICATION: The device may not detect a capture event.

WORKAROUND: The firmware can be coded so that the PCA timer never overflows, thus setting the PCA capture flag.

Another option would be to enable the PCA timer overflow (CF) in the CCON SFR. To do this, clear the CCAPxH and CCAPxL registers to values other than FF00h. Upon an overflow of the PCA timer, the firmware checks the CCAPxH and CCAPxL registers for any new values. If there is a value of FF00h in the SFR pair, CCAPxH/CCAPxL, a capture event has occurred. Clear both registers to values other than FFh and 00h, respectively, for the next capture event.

STATUS: Fix. Refer to Summary Table of Changes for affected steppings.

9804001. Interrupts of 3 or More Priority Levels Occuring at the Same Time

PROBLEM: When three interrupts of different priorities occur at approximately the same time (~40 nsec. window) on the 8x930 peripheral controllers, the lowest priority interrupt will not be serviced correctly. As the lower priority interrupt is interrupted by a higher priority interrupt, the internal interrupt pending bit remains set and the device believes it is still servicing the lowest priority interrupt. This disables any further lowest priority interrupts from being serviced.

Any interrupt source can be affected by this if it is the lowest priority of the three enabled interrupts. For example, if external interrupt 0 (INT0) is priority 3, start of frame (SOF) interrupt is priority 2, and serial port (SIO) interrupt is priority 0, and they all occur at the same time, then the SIO interrupt would stop being serviced.

If four different interrupt priority levels are used, then both priority 0 and 1 interrupts would stop being serviced.

IMPLICATION: This will cause the lowest priority interrupt to no longer be serviced and the interrupt will not recover unless a device reset occurs.

WORKAROUND 1: To insure this interrupt priority conflict does not occur, make sure only two interrupt priority levels are enabled at any given time.

WORKAROUND 2: Once the device is in the failing condition and the lower priority interrupt(s) are no longer being serviced, there is a possible software workaround for some applications. This workaround involves detecting the fail event, manually pushing a return address on the stack, and executing a RETI instruction. The user will have to evaluate if this workaround will apply to their application.

Example:

```
main_code:
      ljmp int_fail_fix
org FF1500H
      Continue main code
      .
      .
      .
int_fail_fix:
      push #00H
      push #15H
      reti
```

How it Works:

The users main program code (main_code) detects the failing condition. One method of doing this would be reading the appropriate interrupt pending bit and starting a timer when this bit was set. If the interrupt is not serviced in a specified amount of time, the program would jump to the int_fail_fix routine. The int_fail_fix routine pushes the address of the instruction after the ljmp to int_fail_fix, 1500 in this case. The RETI instruction will return the program counter to 1500. This will pop the internal interrupt stack so that it will handle future lowest priority interrupt(s).

STATUS: Fix. Refer to Summary Table of Changes for affected steppings.

9806001. 8x930 Cold Boot Issue On Some Hosts

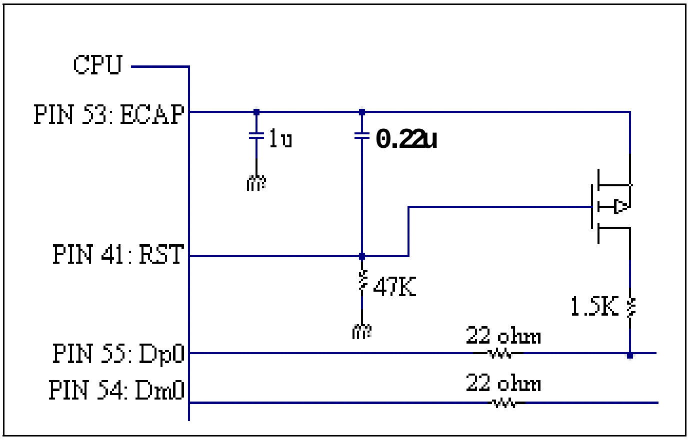
PROBLEM: The problem was first seen when a USB peripheral using the Intel® 8x930 chip was plugged into the USB port of some Toshiba laptops failed to enumerate correctly during a cold boot sequence. The root cause can be tied to the system determining device speed too quickly by misinterpreting power-on transients on the Dm0 line as an indication that the device was low speed when it was really high speed. Some systems may not exhibit this behavior if they power on the USB bus prior to OS initialization thus effectively "hiding" the power-on transients from the host controller/driver. Also, systems that have a host controller/driver that is designed to wait a long enough period of time following power-up prior to device speed determination will not experience this problem (such as systems using the Intel® PIIX4). The problem has been seen on the Toshiba* 500 series and 300 series notebooks (which use a non-Intel host controller) running Microsoft* Windows95/98* with Microsoft* supplied OHCI host controller drivers. The Toshiba* Tecra780* uses the Intel® PIIX4 host controller and does not exhibit this problem. The failure mode at Dp0/Dm0 can be identified by observing a low-speed J at a full-speed device following initial USB reset. Please see the latest USB specification for bus state definitions.

IMPLICATION: USB peripherals using the Intel® 8x930 that are plugged into a system where USB bus power is turned on after OS/host controller initialization (and the system's host controller/host controller driver does not wait long enough prior to speed determination) will not enumerate during a cold boot sequence. Although this problem has only been observed with a full-speed peripheral using the 8x930Ax, it could also occur with other peripherals including low speed peripherals using the 8x930Ax.

WORKAROUND: (no design impact) For peripherals, the user may unplug and re-plug the peripheral once the OS is fully initialized.

WORKAROUND: (design impact) Insert a MOSFET on the Dp0 line delaying the energizing of the pull-up resistor by about 8mS. The circuit inserts a 5 to 8 millisecond delay in powering Dp0 thus eliminating the SE1 caused by power-on transients during the power-on reset of the 8x930. See Figure 3 for details.

Figure 3. Delay Circuitry



The power up transients do not cause failure during a hot attach (attach to a powered port) because the power lines make physical contact before the signal lines (Dm0/Dp0 connectors are 1mm shorter than 5V and GND). The time between the two contacts allows the transients to die down before the system evaluates the speed. Similarly for devices plugged in prior to cold boot, host controllers/drivers should be designed to wait a specified period of time before attempting to evaluate speed to allow for power-on stabilization; please see the latest USB specification for details.

STATUS: There is currently no fix for this problem. Both Toshiba* and Microsoft* have been notified of this issue.

SPECIFICATION CHANGES

001. V_{OH} When I_{OH} = -60 µA

ITEM: The V_{OH} specification given in the DC Characteristics section of the *8x930Ax Universal Serial Bus Peripheral Controller Datasheet* is changed to V_{OH} = {min} V_{CC} – 1.7 V when I_{OH} = -60 µA.

002. Extended Data Float Option

ITEM: Default timings and extended data float timings for the A3 and A4 steppings of the 8*x*930A*x* are provided in Table 3 on page 36 and Table 4 on page 38.

003. AC Characteristics Changed

ITEM: AC characteristics have changed for the A3 and A4 steppings of the 8*x*930A*x* in "Compatibility Mode." The differences between the new characteristics and the AC characteristics for the A2 stepping are summarized in Table 1.

004. Series Resistor Requirement for Impedance Matching Changed to 22 Ohms (no longer 27 – 33 ohms)

ITEM: Section 7.4 of the *8X930Ax Universal Serial Bus Peripheral Controller* data sheet has been changed. For improved signal fidelity, the recommended resistence is 22 ohms.

005. Minimum Operating Voltage (Vcc) Specification for the 8x930Ax

ITEM: Minimum operating voltage changed from 4.0v to 4.15v to better reflect actual testing conditions.

006. 8x930xx Devices No Longer Supported for Low Speed

ITEM: 8x930Ax and 8x930Hx devices are supported at USB hi-speed specification only. This change applies to all 930 Ax/Hx steppings.

Table 1. Summary of 8*x*930A*x* AC Characteristics Changes

Symbol	Parameter	8*x*930A*x* A2 (ns) (1)	8*x*930A*x* A3/A4 (ns) "Compatibility Mode" (EDF# = 1) (2)
T_{AVLL}	Address Valid after ALE Low	(0.5+M) T_{CLK} – 15 [Min]	(0.5+M) T_{CLK} – 13 [Min]
T_{LLAX}	Address Hold after ALE Low	4 [Min] (3)	10 [Min]
T_{WLWH}	WR# Pulse Width	(1+N) T_{CLK} – 12 [Min]	(1+N) T_{CLK} – 10 [Min]
T_{LLRL}	ALE Low to RD# or PSEN# Low	8 [Min]	10 [Min]
T_{LHAX}	ALE High to Address Hold	(1+M) T_{CLK} – 43 [Min]	(1+M) T_{CLK} – 27 [Min]
T_{RLDV}	RD# or PSEN# Low to Valid Data/Instruction In	(1+N) T_{CLK} – 33 [Max]	(1+N) T_{CLK} – 30 [Max]
T_{RLAZ}	RD# or PSEN# Low to Address Float	0 [Max]	3 [Max] (4)
T_{RHDZ2}	Data Float After PSEN# or RD# High	T_{CLK} [Max]	T_{CLK} +10 [Max]
T_{RHLH2}	RD# or PSEN# High to ALE High (Data)	T_{CLK} [Min]	T_{CLK} +10 [Min]
T_{WHLH}	WR# High to ALE High	T_{CLK} + 5 [Min]	T_{CLK} +10 [Min]
T_{AVDV2}	Address (Demuxed) to Valid Data/Instruction In	(2+M+N) T_{CLK} – 48 [Max]	(2+M+N) T_{CLK} – 38 [Max]
T_{AVRL}	Address Valid to RD# or PSEN# Low	(1+M) T_{CLK} – 43 [Min]	(1+M) T_{CLK} – 40 [Min]
T_{AVWL1}	Address (Muxed) Valid to WR# Low	(1+M) T_{CLK} – 43 [Min]	(1+M) T_{CLK} – 40 [Min]

NOTES:

1. Worst-case numbers based on silicon data collected to date.
2. Device configured with default data float timing for fast memory interface
3. At 50° C, T_{LLAX} is 8 ns.
4. Typical value is 0 ns.

SPECIFICATION CLARIFICATIONS

001. T_{RHDZ1} Timing

PROBLEM: The T_{RHDZ1} (Instruction Float After RD#/PSEN# High) specification on the 8*x*930A*x* when operating at 12 MHz is 10 ns. However, a slow memory device such as an EPROM typically takes approximately 30 – 90 ns to float its output. (This specification, generically called T_{PHZ}, may vary depending on the memory type and manufacturer.) Figure 4 and Figure 5 illustrate the T_{RHDZ1} timing.

The difference between the T_{RHDZ1} and T_{PHZ} specifications causes contention on the data bus (P0 in nonpage page, P2 in page mode). The 8X930A***x*** begins to drive the address for the next bus cycle while the memory device is still driving data from the previous bus cycle.

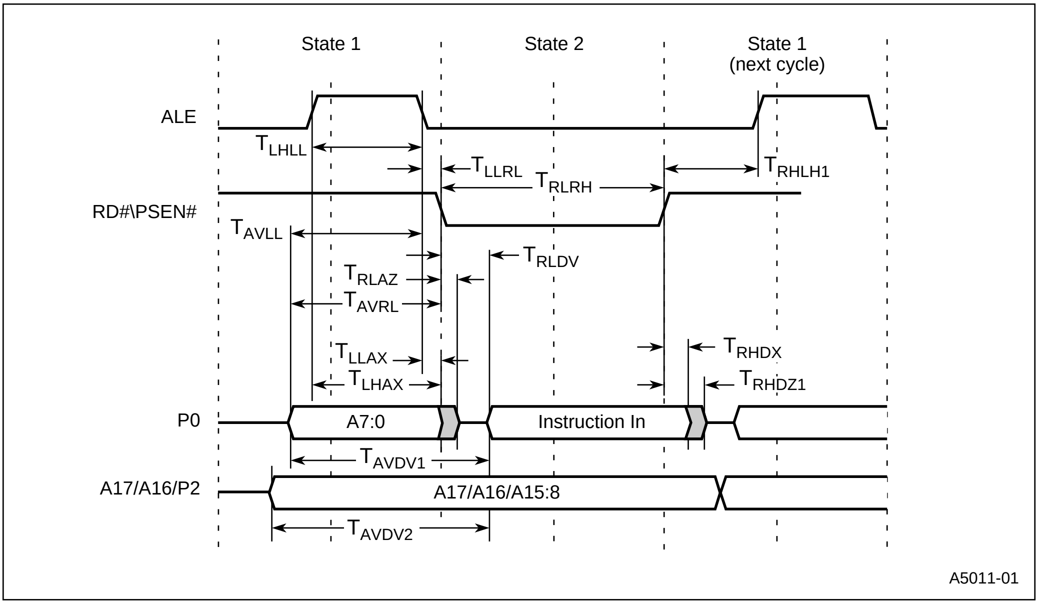


Figure 4. External Code Fetch, Nonpage Mode

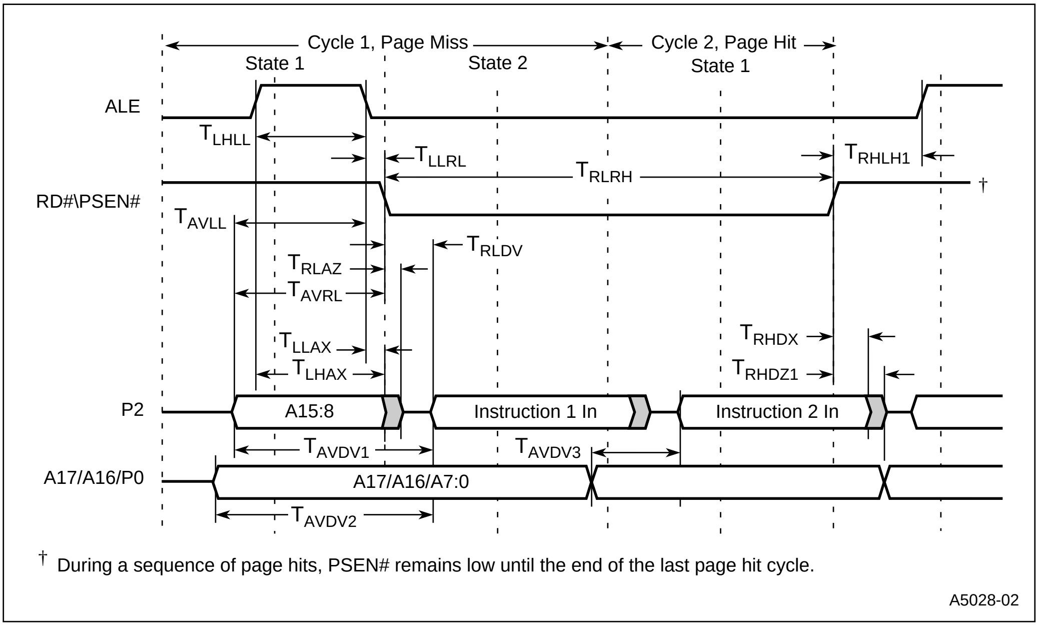


Figure 5. External Code Fetch, Page Mode

To prevent this contention, designers can use a buffer to isolate the output of the memory device from the data bus (port 0 or port 2) of the 8*x*930A*x*. This will prevent the memory device from driving the data bus during the critical period after T_{RHDZ1} expires. We suggest a buffer such as the 74F541 octal, three-state line driver shown in Figure 6.

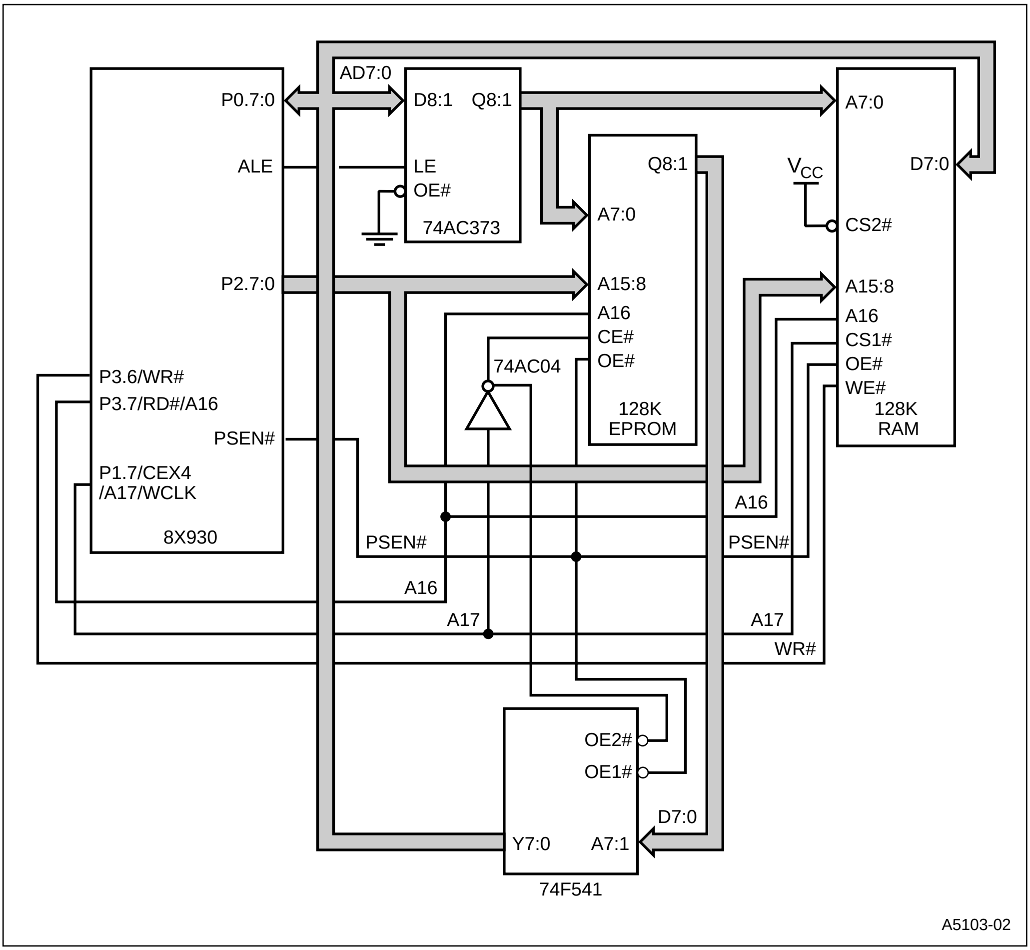


Figure 6. Example Bus Contention Solution for 8X930Ax (Nonpage Mode)

Figure 6 illustrates the connections of an 8X930Ax configured for nonpage mode with EPROM and RAM in external memory. If your system uses a different configuration, your circuit will be different from the example. The 74F541 is enabled to pass data from the EPROM to the 8X930Ax (port 0) when OE1# and OE2# are active. Table is a truth table for the 74F541.

Table 2. Truth Table for 74F541

Inputs			Outputs
OE1#	**OE2#**	**A7:1**	**Y7:1**
L	L	L	L
L	L	H	H
X	H	X	Z
H	X	X	Z

During a read, PSEN# turns the buffer on (OE1# and OE2# are active), connecting the EPROM's output to the 8X930Ax's port 0.

When PSEN# goes high after a read, OE1# and OE2# are deasserted, and the buffer's output switches to the high-impedance state in approximately 9.5 ns (T_{PHZ} for a typical 74F541; the value may vary from one manufacturer to another). Thus, contention on the data bus is prevented. This or a similar hardware solution is recommended for 8X930Ax designs in which memory devices do not meet the T_{RHDZ1} timing specification.

002. Extended Data Float Option

PROBLEM: The T_{RHDZ1} (Instruction Float After RD#/PSEN# High) specification on the 8*x*930A*x* when operating at 12 MHz is 10 ns. However, a slow memory device such as an EPROM typically takes approximately 30 – 90 ns to float its output. (This specification, generically called TPHZ, may vary depending on the memory type and manufacturer.)

IMPLICATION: The difference between the T_{RHDZ1} and T_{PHZ} specifications causes contention on the data bus (P0 in nonpage mode, P2 in page mode). The 8*x*930A*x* begins to drive the address for the next bus cycle while the memory device is still driving data from the previous bus cycle.

WORKAROUND: In addition to the workaround involving the use of tristate buffers, as described in Specification Clarification 001, Intel has modified the 8*x*930A*x* to provide another solution.

In the A-3 stepping of the 8*x*930A*x*, Intel has added an option that allows system designers to increase the value of T_{RHDZ1}. This option is controlled by bit 3 of configuration byte UCONFIG1. This new bit is called the extended data float bit (EDF#). When EDF# is set (1), the controller behaves according to the existing specifications. When EDF# is cleared (0), extended data float timings are in effect and new bus timing specifications apply to the 8*x*930A*x*. The affected parameters are shown in Table 3 and Table 4.

Table 3. 8*x*930A*x* Default and Extended Data Float Timings

Symbol	Parameter	Default Data Float Timing (EDF# =1) "Compatibility Mode" (ns) (1) (2) (4) (5)	Extended Data Float Timing (EDF#=0) "Increased T_{RHDZ1} mode" (ns) (1) (3) (4) (5)
T_{LLAX}	Address Hold after ALE Low	10 [Min]	20 [Min]
T_{RLRH}	RD# or PSEN# Pulse Width	(1+N) T_{CLK} – 10 [Min]	(1+N) T_{CLK} – 32 [Min]
T_{WLWH}	WR# Pulse Width	(1+N) T_{CLK} – 10 [Min]	(1+N) T_{CLK} – 32 [Min]
T_{LLRL}	ALE Low to RD# or PSEN# Low	10 [Min]	20 [Min]
T_{LHAX}	ALE High to Address Hold	(1+M) T_{CLK} – 27 [Min]	(0.5+M) T_{CLK} + 15 [Min]
T_{RLDV}	RD# or PSEN# Low to Valid Data/Instruction In	(1+N) T_{CLK} – 30 [Max]	(1+N) T_{CLK} – 50 [Max]
T_{RHDZ1}	Instruct. Float after PSEN# or RD# High	10 [Max]	(0.5)T_{CLK} – 5 [Max]
T_{RHDZ2}	Data Float After PSEN# or RD# High	T_{CLK} + 10 [Max]	1.5 T_{CLK} – 5 [Max]
T_{RHLH1}	RD# or PSEN# High to ALE High (Instruction)	10 [Min]	0.5 T_{CLK} – 7 [Min]

NOTES:

1. Worst-case numbers based on silicon data collected to date.
2. Device configured with default data float timing for fast memory interface
3. Device configured with extended data float timing for slow memory interface.
4. Values listed in this table are for F_{CLK} = 12 MHz. For F_{CLK} = 6 MHz, T_{CLK} will double to equal 166.6 ns.
5. M = 0,1 is the extended ALE state; N = 0,1,2,3 is the RD#/PSEN#/WR# wait state.

Table 3. 8*x*930A*x* Default and Extended Data Float Timings (Continued)

Symbol	Parameter	Default Data Float Timing (EDF# =1) "Compatibility Mode" (ns) (1) (2) (4) (5)	Extended Data Float Timing (EDF#=0) "Increased T_{RHDZ1} mode" (ns) (1) (3) (4) (5)
T_{RHLH2}	RD# or PSEN# High to ALE High (Data)	T_{CLK} +10 [Min]	1.5 T_{CLK} – 7 [Min]
T_{WHLH}	WR# High to ALE High	T_{CLK} + 10 [Min]	1.5 T_{CLK} – 7 [Min]
T_{AVDV1}	Address (Muxed) Valid to Valid Data/Instruction In	(2+M+N) T_{CLK} – 60 [Max]	(1.5+M+N) T_{CLK} – 28 [Max]
T_{AVRL}	Address Valid to RD# or PSEN# Low	(1+M) T_{CLK} – 40 [Min]	(0.5+M) T_{CLK} + 10 [Min]
T_{AVWL1}	Address (Muxed) Valid to WR# Low	(1+M) T_{CLK} – 40 [Min]	(0.5+M) T_{CLK} + 10 [Min]
T_{AVWL2}	Address (Demuxed) Valid to WR# Low	(1+M) T_{CLK} – 17 [Min]	(1+M) T_{CLK} + 10 [Min]

NOTES:

1. Worst-case numbers based on silicon data collected to date.
2. Device configured with default data float timing for fast memory interface
3. Device configured with extended data float timing for slow memory interface.
4. Values listed in this table are for F_{CLK} = 12 MHz. For F_{CLK} = 6 MHz, T_{CLK} will double to equal 166.6 ns.
5. M = 0,1 is the extended ALE state; N = 0,1,2,3 is the RD#/PSEN#/WR# wait state.

Table 4. 8*x*930A*x* Real-time Wait AC Timing Specifications

Symbol	Parameter	F_{CLK} Variable (ns) Default Data Float Timing (EDF# =1) "Compatibility Mode"			F_{CLK} Variable (ns) Extended Data Float Timing (EDF#=0) "Increased T_{RHDZ1} mode"		
		Min	Typ	Max	Min	Typ	Max
T_{RLYV}	RD# or PSEN# Low to Wait Setup	0		0.5 T_{CLK} – 13	0		0.5 T_{CLK} – 35
T_{WLYV}	WR# Low to Wait Setup	0		0.5 T_{CLK} – 13			0.5 T_{CLK} – 35

003. ECAP Pin Usage to Supply 3.0v to 3.6v Voltage for 1.5K Ohm USB Pullup Terminator

ITEM: For a self-powered or bus-powered device, when the voltage at the V_{CC} pins are at 5.25 V to 4.15V, the voltage at ECAP pin will be at approximately 3.6 V to 3.0V. If the V_{CC} pin is at 4.65 V [Min, Vbus Powered (host or hub) Port specification], the voltage at the ECAP pin will be at approximately 3.2 V (refer to Table 5 below). The capability for this pin to supply the 3.0 V to 3.6 V voltage to the 1.5 K Ohm USB pullup terminator depends upon the V_{CC} voltage level.

For a bus-powered device that is connected to a bus-powered hub, when the voltage at the V_{CC} pins (in the bus-powered devices) are at 4.28v, the voltage at ECAP pin will be at approximately 3.0v. If the V_{CC} voltage drops below 4.28v, the ECAP pin can not supply voltage above 3.0 v for the 1.5K Ohm USB pullup terminator.

NOTE:
The typical ECAP values, listed in the table below, reflect a 1 µF capacitor connection between the ECAP pin and ground.

Table 5. Vcc and Typical ECAP Voltages

V_{CC}	ECAP Pin
5.25v	3.6v
5.00v	3.5v
4.65v	3.2v
4.40v	3.1v
4.28v	3.0v

004. TXCNT Must Be Written With Correct Byte Count

PROBLEM: Both TXCNTH and TXCNTL transmit count registers of the 8*x*930A*x* embedded function transmit endpoint 1 must be written with the correct byte count after moving data into the transmit FIFO data register (TXDAT). When transmitting less than 256 bytes (FFh), the user must initially write TXCNTH with ‘00h’, and then set TXCNTL with the correct transmit byte count.

The 8x930Ax, 8x930Hx User's Manual (order number 272949-001) page numbers 7-17, 7-19 and C-69 contain incorrect descriptions for the reset state of the TXCNTH register, Endpoint 1. The reset state of both TXCNTH and TXCNTL registers are indeterminate.

DOCUMENTATION CHANGES

003. Nonvolatile Memory Verification Port Labeled Incorrectly

ITEM: The illustration “Setup for Verifying Nonvolatile Memory” (Figure 17-1 of the ***8x930Ax, 8x930Hx Universal Serial Bus Microcontroller User’s Manual***) incorrectly depicts P1 as the Verify Modes port. The correct port for Verify Modes is P0.

004. Incorrect Address Given for TXSTAT SFR

ITEM: The USB Function SFR tables (Tables 3-11 and C-7) in the ***8x930Ax, 8x930Hx Universal Serial Bus Microcontroller User’s Manual*** give an incorrect address for the TXSTAT SFR. The correct address is S:F2H.

005. Incorrect Signature Byte

ITEM: Section 17-6 of the ***8x930Ax, 8x930Hx Universal Serial Bus Microcontroller User’s Manual*** mention a signature byte at 61H. There is no signature byte at 61H.

006. Power-on Reset Capacitor Value Changed

ITEM: Figure 14-1 of the ***8x930Ax, 8x930Hx Universal Serial Bus Microcontroller User’s Manual*** depict a 1 µF power-on reset capacitor. The correct value for this capacitor is 0.3 µF.

007. SCON SFR’s REN Bit Description Incorrect

ITEM: The description for the Serial Port Control SFR’s REN bit (SCON.4), as given in Figure 13-2 of the ***8x930Ax, 8x930Hx Universal Serial Bus Microcontroller User’s Manual***, is incorrect. The text should say ”To enable reception, set this bit. To disable reception, clear this bit.”

The SCON SFR also appears in Appendix C of the same manual.

008. Extraneous Footnote in RXCON SFR

ITEM: The dagger footnote (†) does not apply to the RXFFRC and RXISO bits in the RXCON SFR, as shown in Figure 7-15 of the ***8x930Ax, 8x930Hx Universal Serial Bus Microcontroller User’s Manual***. The RXCON SFR also appears in Appendix C. Note that the dagger footnote ***does*** apply to the SFR’s ADVWM and REVWP bits.

009. Power Off Flag Voltage Values

ITEM: Section 15.2.2 of the *8x930Ax, 8x930Hx Universal Serial Bus Microcontroller User's Manual* states that "the hardware sets the Power Off Flag (POF) in PCON when V_{CC} rises from < 3 V to > 3 V to indicate that on-chip volatile memory is indeterminate...since for V_{CC} < 3 V data may have been lost or some logic may have malfunctioned." The voltage value should be 3.5 V for all references, not 3 V.

010. W_{CLK} Description Incorrect

ITEM: The description for the wait clock output (W_{CLK}) given in Table 16-1 and Table B-2 of the *8x930Ax, 8x930Hx Universal Serial Bus Microcontroller User's Manual* is incorrect. Instead of "When enabled, the W_{CLK} output produces a square wave signal with a period of one-half the oscillator frequency," the final sentence should read "When enabled, the W_{CLK} output produces a square wave signal with a period of T_{CLK}."

011. Configuration Byte Misspelled

ITEM: On page 16-11 of the *8x930Ax, 8x930Hx Universal Serial Bus Microcontroller User's Manual*, in the third paragraph of the note, UNCONFIG0 should be UCONFIG0.

012. RTWCE Description Inaccurate

ITEM: On page 16-11 of the *8x930Ax, 8x930Hx Universal Serial Bus Microcontroller User's Manual*, in Figure 16-11, the description of RTWCE should say "with RTWE set, setting RTWCE will enable the WAIT clock....." In other words, setting RTWCE alone will not enable the wait clock, both bits must be set.

013. RL Instruction Misspelled

ITEM: On page A-4 of the *8x930Ax, 8x930Hx Universal Serial Bus Microcontroller User's Manual*, the 'A' should be moved to the second line for RLA and RLCA. The instruction is RL A not RLA.

014. Footnote Incorrect in Data Instructions Table

ITEM: On page A-6 of the *8x930Ax, 8x930Hx Universal Serial Bus Microcontroller User's Manual*, Note 2 for Table A-9 is not correct. ORL, ANL, and XRL all have one instruction that uses DRk (see page A-38 for an example).

015. Capacitor Value Change in Figure 3 of this Specification Update

ITEM: The capacitor value in Figure 3. Delay Circuitry, in Errata 9806001 of this specification update has changed. The capacitor value is 0.22µ, not 0.23µ.